Applied Mathematics and Computational Intelligence
Volume 13, No. 1, Feb 2024 [26-35]

Mitigating Overfitting in Extreme Learning Machine Classifier Through Dropout Regularization

F K Q Alnagashi[1], N Abdul Rahim[1,2,*], S A Abdul Shukor[1,2], M H A Hamid[1,2]

[1]Faculty of Electrical Engineering & Technology, Universiti Malaysia Perlis, 02600 Arau, Perlis, Malaysia
[2]Centre of Excellence for Intelligent Robotics & Autonomous System (CIRAS), Universiti Malaysia Perlis, 02600 Arau, Perlis, Malaysia

[*] Corresponding author: norasmadi@unimap.edu.my

Received: 16 January 2024; Revised: 25 January 2024; Accepted: 29 January 2024; Published: 29 February 2024

ABSTRACT

Achieving optimal machine learning model performance is often hindered by the limited availability of diverse datasets, a challenge exacerbated by small sample sizes in real-world scenarios. In this study, we address this critical issue in classification tasks by integrating the Dropout technique into the Extreme Learning Machine (ELM) classifier. Our research underscores the effectiveness of Dropout-ELM in mitigating overfitting, especially when data is scarce, leading to enhanced generalization capabilities. Through extensive experiments on synthetic and real-world datasets, our findings consistently demonstrate that Dropout-ELM outperforms traditional ELM, yielding significant accuracy improvements ranging from 0.19% to 16.20%. By strategically implementing dropout during training, we promote the development of robust models that reduce reliance on specific features or neurons, resulting in increased adaptability and resilience across diverse datasets. Ultimately, Dropout-ELM emerges as a potent tool to counter overfitting and bolster the performance of ELM-based classifiers, particularly in scenarios with limited data. Its established efficacy positions it as a valuable asset for enhancing the reliability and generalization of machine learning models, providing a robust solution to the challenges posed by constrained training data.

Keywords: Artificial datasets**,** Classification**,** Dropout**,** Machine Learning**,** Real-world datasets, small sample-sized**,** Test Accuracy

1 INTRODUCTION

Classification Machine learning has made significant strides in solving complex problems across diverse domains [1]. However, the efficacy of machine learning models is highly contingent on the availability of ample and diverse datasets. In practical scenarios, researchers often encounter limited data availability, posing challenges that necessitate innovative solutions for building accurate and robust models [2]. This paper addresses the application of Dropout, a regularization technique, to enhance the Extreme Learning Machine (ELM) classifier's performance under the constraints of small sample size data. The realm of machine learning has traditionally thrived on vast datasets, enabling models to learn intricate patterns and representations effectively. In a classification of information, people will deal with two types of data, either huge or small samples. However, small sample-sized (SSS) data are the most dealt with, especially in the medical and healthcare field. When confronted with a notably restricted sample size, particularly in the context of Single Sample Size (SSS) data and employing a solitary classifier, the tendency is for machine-learning algorithms to undergo undertraining, resulting in their ineffectiveness. In certain instances, within SSS problems, particularly extreme ones, a scenario may arise where a substantial ratio exists between the number of features and the number of instances. In such situations, the sheer volume of features in comparison to the limited instances can lead to overfitting of the classification algorithm.

The utilization of small sample size data introduces a series of significant challenges, each posing considerable obstacles:

i) Sample Representativeness: Small datasets may inadequately capture the full complexity and heterogeneity of the underlying data distribution, limiting the model's ability to generalize effectively [3].

ii) Overfitting and Variance: In the presence of limited samples, models tend to overfit and capture noise, leading to increased variance and poor generalization to unseen data [4].

iii) High Dimensionality: High-dimensional feature spaces aggravate the curse of dimensionality in small datasets, exacerbating sparsity issues and hindering the model's ability to discern salient features [5].

2 INTRODUCTION TO DROPOUT AS A REGULARIZATION TECHNIQUE FOR THE ELM CLASSIFIER

Dropout, initially introduced as a regularization method for neural networks, has since gained widespread adoption across various machine learning architectures. The ELM classifier, renowned for its simplicity and computational efficiency, stands to benefit from the Dropout technique, especially when confronted with limited training samples. Dropout operates by randomly deactivating neurons during training, effectively encouraging the model to learn redundant representations and reducing interdependencies among neurons. This regularization process enhances the model's robustness and resilience to overfitting, making it a compelling choice for addressing the challenges posed by small sample size data [6]. By exploring the intersection of Dropout regularization and the ELM classifier in the context of small sample size data, this paper aims to provide valuable insights into the practical efficacy of this approach. Subsequent sections will delve into the theoretical underpinnings, experimental setup, and empirical results, contributing to a comprehensive understanding of the presented regularization technique's applicability and impact on the ELM classifier's performance.

3 BACKGROUND AND RELATED WORK

3.1 Overview of Extreme Learning Machine (ELM)

The extreme learning machine (ELM) is a quick and effective training procedure for single hidden layer feed-forward neural networks (SLFNs). It has been effectively used in a variety of fields, such as pattern recognition, machine vision, and the processing of biological data. The ELM algorithm's structure has a strong impact on how well it performs. In the Extreme Learning Machine (ELM), the number of neurons in the hidden layer and the input neuron matrix are specified before a random pattern is used to generate the input weight matrix. The sample matrix's size affects how many input neurons are needed, whereas the number of hidden layer neurons is selected manually. As a result, the size of the sample matrix may affect how well ELM works. The number of hidden layer neurons needs to be carefully tuned for ELM to function optimally without affecting the sample size. When the training sample closely resembles the input weight matrix created by a random pattern, ELM's accuracy in both regression and classification tasks can be significantly increased. The ELM classifier has garnered significant attention due to its inherent advantages in handling small sample-size data. Lixin Zheng et al. [7] elucidated the unique characteristics of ELM, emphasizing its computational efficiency and non-iterative learning process. They demonstrated the superior generalization performance of ELM on small datasets, making it a promising candidate for applications with limited training samples.

Studying neurons' numbers in hidden layers also fine-tuning the parameters of the input weight matrix are important steps in boosting the effectiveness of the Extreme Learning Machine (ELM) algorithm [8]. Although ELM has good generalization and fast performance, there is still room for improvement for example, when randomly assigning parameters. To address this issue and improve performance with more efficient networks, the concept of E-ELM (Enhanced Extreme Learning Machine) was introduced. E-ELM aims to eliminate redundancy among hidden nodes and create more compact networks, leading to improved performance [9, 10]. ELM has shown superiority over support vector machines in classifying microarray data[11]. However, neural network classifiers are prone to over-training, which can lead to a decrease in generalization performance. To tackle this challenge, Lan et al. [12] proposed an approach to overcome it by integrating multiple neural networks, specifically extending the ELM algorithm through the averaging of outputs from individual classifiers in small sample-sized classification problems, such as those in the medical and healthcare fields, a single classifier may overfit the data resulting in a complex model, so using multiple classifiers can be beneficial.

4 DROPOUT METHOD

4.1 Dropout and Its Application to the ELM Classifier

Dropout is a powerful regularization technique originally developed for neural networks, but its versatility has been demonstrated across various machine learning models, including the (ELM classifier). Dropout operates by randomly deactivating neurons during the training process. In the context of the ELM classifier, Dropout is integrated into the hidden layer neurons. During each forward pass of training, individual neurons are stochastically dropped out with a certain probability, while during the backward pass, only active neurons receive gradients [6].

The application of Dropout to the ELM classifier is particularly beneficial in scenarios with small sample size data. By introducing random dropout during training, the model is encouraged to learn more robust and independent representations. The dropout process acts as a form of model averaging, effectively preventing co-adaptation of neurons and promoting better generalization on unseen data. This regularization technique enables the ELM classifier to become less reliant on specific neurons, thereby mitigating the risk of overfitting and improving its performance on limited training samples.

The Dropout algorithm is conceptually straightforward. During the forward pass of training, each neuron in the hidden layer is retained with a probability (dropout rate) p and deactivated (set to zero) with a probability of (1 - p). This dropout process is independently applied to each neuron, ensuring that the activations are stochastically masked during training. In the backward pass, only the active neurons receive gradients, while the deactivated neurons remain unaffected. The Dropout technique effectively simulates an ensemble of exponentially many thinned networks during training. Consequently, the model learns to adapt to various sub-networks, effectively averaging out their predictions during inference. This ensemble effect fosters improved generalization, particularly crucial in small sample-size data settings. The ability of Dropout to address overfitting in small sample size data stems from its capacity to reduce co-dependencies among neurons [13]. In conventional deep learning settings, neurons can become strongly interdependent, leading to overfitting and limited generalization. By randomly dropping neurons during training, Dropout prevents the model from relying too heavily on specific connections, encouraging the exploration of alternative pathways. This exploration aids in discovering more diverse and robust features, better capturing the underlying data distribution, and mitigating the risk of overfitting.

5 MODEL DESCRIPTION

This section elucidates the concept of the dropout neural network model. Consider a neural network comprising L hidden layers. In this context, the notation Let $l \in \{1,\ldots, L\}$ designates the hidden layers within the network. The vector of inputs into layer l is represented as$z^{(l)}$, while the vector of outputs from layer l is denoted as $y^{(l)}$ (where $y^{(0)} = x$, representing the input layer). The weights and biases at layer l are respectively denoted as $W^{(l)}$ and $b^{(l)}$.

The forward propagation process of a neural network can be succinctly expressed as follows: for any layer index l (where l lies in the range of $\{0,\ldots,L-1\}$), and for each hidden unit i the corresponding operations are carried out.

$$z_i^{(l+1)} = W^{(l+1)} y^l + b_i^{(l+1)} \tag{1}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \tag{2}$$

$$r_j^l \sim \text{Bernoulli}(\text{p}), \tag{3}$$

$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)} \tag{4}$$

In this context, $r^{(l)}$ represents a vector consisting of Bernoulli random variables. Each of these variables carries a probability p of being equal to 1. This vector is sampled individually for each layer. It is then element-wise multiplied with the corresponding outputs of that layer, denoted as $y^{(l)}$, resulting in the creation of thinned outputs $y^{(l)}$. Subsequently, these thinned outputs serve as inputs for the subsequent layer in the network. During the learning process, the derivatives of the loss function are backpropagated through this thinned network structure. At the point of testing, a scaling operation is applied to the weights: $W^{(l)} = pW^{(l)}$. This modified neural network, which incorporates the scaled weights, is then executed without the dropout mechanism in place.

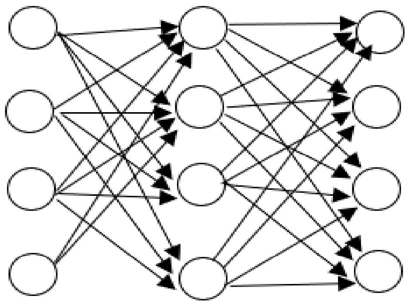
Figure 1: Standard neural network

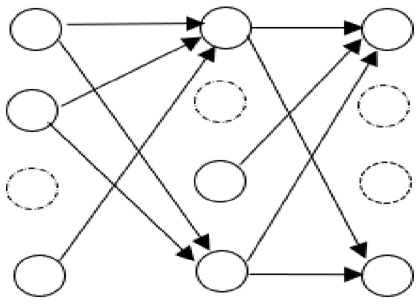
Figure 2: After applying dropout

6 EXPERIMENTAL SETUP: DESCRIPTION OF THE GENERATED DATASET FOR EVALUATION

All evaluations were carried out in Python / Anaconda 2.2.0, running on a desktop with a 2.40 GHz CPU, 16 GB RAM, and 500 GB hard disk. following experiments were d-signed to evaluate the performance of Dropout-ELM (it is noted that all the results in this paper are averages of 10 repeated independent experiments.

Experiment 1: Using ELM and ELM-Dropout method for real-world datasets, which have been chosen based on a small sample-sized classification problem for multiclass attributes.

Experiment 2: Using ELM and ELM-Dropout method for Artificial dataset, which has been generated based on a small sample-sized classification problem for multiclass attributes. To comprehensively evaluate the efficacy of the Dropout regularization technique when applied to the Extreme Learning Machine (ELM) classifier, particularly within the constraints of small sample size data, a two-fold approach was adopted. This involved the utilization of both synthetic and real-world datasets. For the synthetic dataset, the Weka application was employed on a laptop to generate controlled data instances. The primary objective behind generating synthetic data was to exercise precise control over the dataset's characteristics. This approach allows for a meticulous examination of the influence of varying feature numbers and instance sizes on the performance of the model under investigation. Furthermore, the inclusion of real-world datasets is a crucial aspect of this evaluation process. By incorporating genuine datasets into the analysis, the applicability and effectiveness of the Dropout regularization technique on the ELM classifier can be assessed in scenarios that more closely mimic real-world conditions. This juxtaposition of synthetic and real data enhances the robustness and relevance of the study's findings. In essence, this hybrid methodology provides a comprehensive evaluation framework, offering insights into how the Dropout regularization technique impacts the performance of the ELM classifier in the context of both controlled synthetic data and authentic real-world data. This approach ensures a balanced and insightful analysis that can contribute significantly to our understanding of the technique's utility in handling small sample size data.

7 DATASET

7.1 Artificial dataset

This experiment starts with the generation of artificial data using WEKA software [14]. Data sets are generated with a built-in data generator in WEKA using the properties shown in Table 1.

Data sets will be generated with a number of features ranging from 50 and 100. Furthermore, for each feature number, 3 data sets will be produced, each with a different number of instances ranging from 100,200, 300, and 400 with an interval of 100, yielding a total of 8 data sets. Finally, the number of Classes is set as 3. The naming, number of instances, number of features, and instance-to-feature ratio (N/M) for data sets were listed in Table 1 as well. By varying the number of features and instances, the generated datasets encompass different levels of data sparsity and complexity. This diversity allows for a comprehensive evaluation of the Dropout-ELM classifier's performance under various conditions and provides valuable insights into its behavior with small sample size data. For clarification, the naming of data sets with 50 features will be prefixed with the alphabet 'A', and 100 features will be prefixed with the alphabet 'B'. The class number will be after the name of the data set with an '_' separator. For example, a data set with the name 'B4_3' will have 3 classes. For both types of datasets used in this research, the number of instances in a data set is N, and the number of features is M. According to Vapnik [15], a data set was considered small when the ratio N/M is less than 20 [16].

Table 1: Naming and properties of c data sets.

Data set	Instance	Feature	Class	N/M
A1_2	100	50	2	2
A2_2	200	50	2	4
A3_2	300	50	2	6
A4_2	400	50	2	8
B1_3	100	100	3	1
B2_3	200	100	3	2
B3_3	300	100	3	3
B4_3	400	100	3	4

7.2 Real-world dataset

In the first stage of this experiment, the real data will be chosen based on a small sample-sized classification problem for multiclass attributes and it will be used to evaluate the new ensemble classifier. For this stage, the real data will be randomly from the Kaggle website [17].

Five data sets will be used for features 4,28,34,64 and 10. Each with a different number of instances which are 150,325,337,768 and 100. The number of classes was 2,4 and 10 classes. Table 2 presents the names, instance counts, feature counts, and the instance-to-feature ratio (N/M) for the datasets.

Table 2: Naming and properties of Real-Word data sets.

NO	Data Set	INSTANCE	FEATURE	CLASS	N/M
1	Iris	150	4	2	18.7
2	Forest	325	28	4	11.60
3	Ionosphere	337	34	2	9.9
4	Pima	768	64	10	12
5	Fertility	100	10	2	10

For the next steps, the generated datasets with their respective characteristics will serve as the foundation for conducting the experimental evaluation of the Dropout-ELM classifier's performance. The model will be trained and tested on these synthetic datasets, and the results will be analysed and compared to determine the effectiveness of Dropout regularization in handling small sample size data in the context of the ELM classifier.

8 EXPERIMENTAL RESULTS AND ANALYSIS

As mentioned in the previous section, there are 2 experiments, the first using artificial data sets, and the other involving real-world data sets. To verify the effectiveness of the proposed algorithm denoted by Dropout-ELM for convenience, experimentally compare the proposed algorithm Dropout-ELM with the original ELM on 13 small data sets. In the 13 data sets, there are 8 artificial data set and 5 real-world data set from the Kaggle website. Our primary objective was to validate the effectiveness of the Dropout-ELM algorithm through comprehensive comparisons. In the experiments, each data set is randomly partitioned into training and testing sets, with 80% of instances used for training and 20% for testing.

8.1 Classification results on artificial data sets

The classification results obtained using the original ELM algorithm and Dropout-ELM on artificial data sets are showcased. The outcomes of this experiment are summarized in Table 3.

Table 3: Experimental results on artificial data sets using normal ELM and Dropout-ELM

		Results on Artificial data sets using Normal ELM			**Results on Artificial data sets using Dropout-ELM**		
Datasets	**Hidden nodes**	**Average Train Accuracy**	**Average Test Accuracy**	**Different**	**Train Accuracy**	**Test Accuracy**	**Different**
A1_2	97	95.50	83.50	12	97.22	87.00	10.22
A2_2	56	92.31	95.00	2.69	93.9	96.00	2.1
A3_2	62	96.62	94.66	1.96	96.81	97.00	0.19
A4_2	74	97.28	94.75	2.53	98.38	99.75	1.37
B1_3	43	94.20	78.00	16.2	92.85	86.33	6.52
B2_3	94	98.42	93.33	5.09	98.00	93.50	4.5
B3_3	79	97.61	94.88	2.73	98.23	96.74	1.49
B4_3	85	97.17	95.16	2.01	98.71	97.66	1.05

8.2 Classification results on Real-World data sets

To assess the impact of the proposed Dropout-ELM algorithm, experiments were conducted on different (Real-World) datasets, and the outcomes of these experiments are summarized in Table 4.

Table 4: Experimental results on Real-world data sets using normal ELM and Dropout-ELM

		Results on Real-world data sets using Normal ELM			**Results on Real-world data sets using Dropout-ELM**		
Data Set	**Hidden nodes**	**Train Accuracy**	**Test Accuracy**	**Different**	**Train Accuracy**	**Test Accuracy**	**Different**
Iris	8	96.51	95.33	1.18	96.50	95.40	1.10
Forest	40	83.45	82.60	0.85	82.85	83.10	0.25
Ionosphere	60	93.53	89.09	4.44	92.22	94.14	1.92
Pima	90	83.51	81.76	1.75	86.63	84.90	1.73
Fertility	15	85.36	82.67	2.96	89.60	90.02	0.42

From Table 3, which shows the result of artificial datasets in ELM and Dropout-ELM, a notable trend emerges. While the Normal ELM algorithm demonstrates impressive train accuracies across various datasets, the test accuracies sometimes fall short, indicative of potential overfitting. In contrast, the Dropout-ELM algorithm consistently achieves competitive test accuracies that are on par with or even surpassing the train accuracies. This alignment suggests that the Dropout-ELM algorithm effectively mitigates overfitting tendencies, resulting in more balanced performance on unseen data. In addition, from the experimental results presented in the tables above, it is evident that the Dropout-ELM algorithm consistently exhibits improvements over the original ELM algorithm. Notably, in various instances, the test accuracy of the Dropout-ELM algorithm surpasses that of the original ELM, indicating its effectiveness in enhancing generalization performance. The utilization of dropout in the ELM framework appears to alleviate overfitting, leading to more reliable and robust results.

9 CONCLUSION

The study focused on assessing the effectiveness of the Dropout-ELM algorithm, an augmentation to the conventional Extreme Learning Machine (ELM). Extensive experiments were conducted across diverse datasets, both artificial and real-world, to investigate the impact of Dropout-ELM on classification performance.

The experiments consistently revealed that Dropout-ELM outperforms the traditional ELM approach by addressing overfitting, a prevalent challenge in machine learning. By introducing dropout during training, Dropout-ELM enhances model generalization. The algorithm's capacity to prevent excessive reliance on specific features or neurons contributes to more adaptable and robust models. The outcomes underscore the potential of Dropout-ELM to enhance test accuracies, indicative of better generalization across datasets, with an overall percent improvement range observed between 0.19% and 16.20%. This improvement aligns with the overarching goal of machine learning deploying models that excel with unseen data. Dropout-ELM's efficacy in mitigating overfitting makes it a valuable technique for bolstering the reliability of ELM-based classification tasks.

ACKNOWLEDGMENT

This research was supported by the Ministry of Higher Education, Malaysia, through the Fundamental Research Grant Scheme (Ref: FRGS/1/2020/ICT03/UNIMAP/02/2). The authors also acknowledge the Faculty of Electrical Engineering & Technology, Universiti Malaysia Perlis (UniMAP) for the equipment and technical assistance.

REFERENCES

[1] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends,perspectives, and prospects,” *Science* (80-.)., vol. 349, no. 6245, pp. 255–260, 2015.

[2] J. Zhai, L. Zang, and Z. Zhou, “Ensemble dropout extreme learning machine via fuzzy integral for data classification,” *Neurocomputing*, vol. 275, pp. 1043–1052, 2018, doi: 10.1016/j.neucom.2017.09.047.

[3] L. Alzubaidi et al., Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, *Springer International Publishing*, vol. 8, no. 1., 2021. doi: 10.1186/s40537-021-00444-8.

[4] X. Ying, “An Overview of Overfitting and its Solutions,” *J. Phys. Conf. Ser.*, vol. 1168, no. 2, 2019, doi: 10.1088/1742-6596/1168/2/022022.

[5] W. Jia, M. Sun, J. Lian, and S. Hou, “Feature dimensionality reduction: a review,” *Complex Intell. Syst.*, vol. 8, no. 3, pp. 2663–2693, 2022, doi: 10.1007/s40747-021-00637-x.

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[7] G. Liang, H. Hong, W. Xie, and L. Zheng, “Combining Convolutional Neural Network With Recursive Neural Network for Blood Cell Image Classification,” *IEEE Access*, vol. 6, no. c, pp. 36188–36197, 2018, doi: 10.1109/ACCESS.2018.2846685.

[8] Z. Zhao and X. Zhang, “Theory and Numerical Analysis of Extreme Learning Machine and Its Application for Different Degrees of Defect Recognition of Hoisting Wire Rope,” *Shock Vib.*, vol. 2018, 2018, doi: 10.1155/2018/4168209.

[9] Q. Y. Zhu, A. K. Qin, P. N. Suganthan, and G. Bin Huang, “Evolutionary extreme learning machine,” *Pattern Recognit.*, vol. 38, no. 10, pp. 1759–1763, 2005, doi: 10.1016/j.patcog.2005.03.028.

[10] J. Cao, Z. Lin, and G. Bin Huang, “Composite function wavelet neural networks with differential evolution and extreme learning machine,” *Neural Process. Lett.*, vol. 33, no. 3, pp. 251–265, 2011, doi: 10.1007/s11063-011-9176-y.

[11] Y. Zhao, G. Wang, Y. Yin, Y. Li, and Z. Wang, “Improving ELM-based microarray data classification by diversified sequence features selection,” *Neural Comput. Appl.*, vol. 27, no. 1, pp. 155–166, 2016, doi: 10.1007/s00521-014-1571-7.

[12] Y. Lan, Y. C. Soh, and G. Bin Huang, “Ensemble of online sequential extreme learning machine,” *Neurocomputing*, vol. 72, no. 13–15, pp. 3391–3395, 2009, doi: 10.1016/j.neucom.2009.02.013.

[13] M. Abdar et al., “A review of uncertainty quantification in deep learning: Techniques, applications and challenges,” *Inf. Fusion*, vol. 76, pp. 243–297, 2021, doi: 10.1016/j.inffus.2021.05.008.

[14] J. M. Alonso and A. Bugarin, “ExpliClas: Automatic Generation of Explanations in Natural Language for Weka Classifiers,” *IEEE Int. Conf. Fuzzy Syst.*, vol. 2019-June, pp. 1–6, 2019, doi: 10.1109/FUZZ-IEEE.2019.8859018.

[15] V. N. Vapnik, “An overview of statistical learning theory,” *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999, doi: 10.1109/72.788640.

[16] N. H. Ruparel and N. M. Shahane, “Learning from Small Data Set to Build Classification Model: A Survey,” *Int. J. Comput. Appl.*, no. 4, 2013.

[17] R. Mohammed, J. Rawashdeh, and M. Abdullah, “Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results,” in 2020 11th Int. Conf. Inf. Commun. Syst. ICICS 2020, pp. 243–248, 2020, doi: 10.1109/ICICS49469.2020.239556.